CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- Before this Amendment: Claims 1-32.
- After this Amendment: Claims 1-32

Non-Elected, Canceled, or Withdrawn claims: none

Amended claims: 1, 14, 21, 27-29 and 32

New claims: none

Claims:

1. (Currently Amended) A programming interface embodied on one or more computer ~~readable~~ storage media, comprising:

~~a first group of functions related to communicating~~ instructions to communicate a new security policy to a plurality of security engines, wherein each of the plurality of security engines is configured to replace an existing security policy with the new security policy, and wherein a new set of rules and/or data associated with the new policy is provided to each security engine; and

~~a second group of functions related to communicating~~ instructions to communicate an indication of each security engine's readiness to implement the new security policy, wherein each of the plurality of security engines returns a value signifying whether it has processed the new set of rules and/or data received to indicate readiness to implement the new security policy.

2. **(Original)** A programming interface as recited in claim 1 wherein the first group of functions includes a method that instructs each of the plurality of security engines to delete the new security policy.

3. **(Original)** A programming interface as recited in claim 1 wherein the first group of functions includes a method that initializes a particular security engine.

4. **(Original)** A programming interface as recited in claim 1 wherein the first group of functions includes a method that instructs each of the plurality of security engines to implement the new security policy.

5. **(Original)** A programming interface as recited in claim 1 wherein the first group of functions further comprises a method that communicates new data associated with an existing security policy to at least one of the plurality of security engines.

6. **(Original)** A programming interface as recited in claim 1 wherein the first group of functions further comprises a method that communicates configuration information to at least one of the plurality of security engines.

7. **(Original)** A programming interface as recited in claim 1 wherein the second group of functions includes a method that indicates whether a particular security engine has implemented the new security policy.

8. **(Original)** A programming interface as recited in claim 1 wherein the second group of functions further comprises a method that retrieves updated data associated with a particular security policy.

9. **(Original)** A programming interface as recited in claim 1 wherein the second group of functions further comprises a method that communicates new data identified by one of the plurality of security engines to a security agent.

10. **(Original)** A programming interface as recited in claim 1 wherein the second group of functions further comprises a method that allows one of the plurality of security engines to query a user of a system containing the plurality of security engines.

11. **(Original)** A programming interface as recited in claim 1 wherein at least one of the plurality of security engines implements an antivirus service.

12. **(Original)** A programming interface as recited in claim 1 wherein at least one of the plurality of security engines implements a firewall application.

13. (Original) A programming interface as recited in claim 1 wherein the plurality of security engines implement the new security policy after all security engines have indicated a readiness to implement the new security policy.

14. (Currently Amended) A computer system including:

one or more microprocessors; and

one or more software programs, the one or more software programs utilizing an application program interface to implement a security policy on a plurality of security engines, the application program interface comprising the following functions:

a first function that communicates a new security policy to the plurality of security engines, wherein a new set of rules and/or data associated with the new policy is communicated;

a second function that identifies whether each of the plurality of security engines is prepared to apply the new security policy based on a value generated by each of the plurality of security engines signifying whether it has processed the new set of rules and/or data; and

a third function that instructs each of the plurality of security engines to implement the new security policy after determining that all of the security engines are prepared to apply the new security policy.

15. (Original) A computer system as recited in claim 14 further comprising a fourth function that causes each of the plurality of security engines to delete the new security policy if at least one of the plurality of security engines is unable to apply the new security policy.

16. (Original) A computer system as recited in claim 14 further comprising a fourth function related to communicating event information identified by a first security engine to the other security engines.

17. (Original) A computer system as recited in claim 14 further comprising a fourth function related to communicating security-related information identified by a first security engine to an event manager.

18. (Original) A computer system as recited in claim 17 wherein the event manager communicates the security-related information to at least one of the plurality of security engines.

19. (Original) A computer system as recited in claim 14 wherein at least one of the plurality of security engines is associated with a first type of security attack.

20. (Original) A computer system as recited in claim 19 wherein at least one of the plurality of security engines is associated with a second type of security attack.

21. (Currently Amended) A method comprising:

calling ~~one or more~~ at least one of a plurality of first functions to facilitate communicating a security policy to a first security engine;

calling ~~one or more~~ at least one of a plurality of second functions to facilitate determining whether the first security engine has applied the security policy; and

calling ~~one or more~~ at least one of a plurality of third functions to facilitate communicating security-related information from the first security engine to a second security engine, wherein the first security engine communicates whether it is ready to apply the security policy.

22. (Original) A method as recited in claim 21 wherein the security-related information identifies a type of security attack.

23. (Original) A method as recited in claim 21 further comprising calling one or more fourth functions to facilitate interacting with a user of a system containing the first security engine.

24. (Original) A method as recited in claim 21 further comprising calling one or more fourth functions to facilitate communicating configuration information to the first security engine.

25. **(Original)** A method as recited in claim 21 further comprising calling one or more fourth functions to facilitate instructing the first security engine and the second security engine to implement the security policy.

26. **(Original)** A method as recited in claim 21 further comprising calling one or more fourth functions to facilitate communicating a revised security policy to the first security engine.

27. **(Currently Amended)** A system comprising:

means for storing instructions facilitating an application program interface implementing a security policy on a plurality of security engines;

means for exposing a first function that communicates a security-related event to an event manager, wherein the communication of the security-related event includes information or details of the event being communicated;

means for exposing a second function that identifies a plurality of security engines associated with the security-related event, wherein the identified security engines are those security engines determined to be able to use the event information ; and

means for exposing a third function that communicates the security-related event from the event manager to the identified security engines thus each of the plurality of security engines need not know of the other security engines;

means for exposing a fourth function that communicates a new security policy from the event manager to the plurality of security engines to increase security based on shared event information;

means for exposing a fifth function that instructs the plurality of security engines to replace an existing security policy with the new security policy; and

means for exposing a sixth function that communicates the ability of the plurality of security engines to replace an existing security policy with the new security policy.

28. (Currently Amended) A system as recited in claim 27 further comprising:

~~means for exposing a fourth function that communicates a new security policy to the plurality of security engines; and~~

~~means for exposing a fifth function that instructs the plurality of security engines to replace an existing security policy with the new security policy~~

means for exposing a seventh function that instructs the plurality of security engines to implement the new security policy if all of the plurality of security engines can implement the new security policy.

29. (Currently Amended) A system as recited in claim 28 further comprising means for exposing a ~~sixth~~ function that instructs the plurality to security engines to delete the new security policy if at least one of the plurality of security engines cannot implement the new security policy.

30. (Original) A system as recited in claim 27 wherein the security-related event is detection of a virus.

31. (Original) A system as recited in claim 27 wherein the security-related event is an unauthorized attempt to access a storage device.

32. (Currently Amended) A system as recited in claim 27 further comprising means for exposing a ~~fourth~~ function that notifies the event manager that a particular security engine has finished processing another function call.